IN THE CLAIMS:

Please amend claims as follows.

1. (original) Method for analysing a set of original description files for an integrated digital electronic system (5) in a description language called the HDL language, at the register transfer level, in order to automatically insert instructions in the HDL language into description files to obtain a new set of description files in the HDL language for the integrated digital electronic system including test functions such that during automatic synthesis of the integrated digital electronic system using a new set of files, the integrated digital electronic system obtained builds in at least some of the logical electronic circuits (22, 23, 24) necessary to test the integrated digital electronic system (5),

method characterised in that it includes the following steps:

- automatic localisation (1) of HDL instruction sequences that will be at the origin of the memory elements during synthesis of the system (3), in the original HDL description files;
- automatic and sequential insertion in at least part of HDL description files, and without relational or functional analysis of the identified memory elements, of so–called SCAN HDL instructions used to obtain at least one so–called "SCAN" chain (11) connecting the memory elements, during synthesis of the system (3).

2. (original) Analysis and insertion method according to claim 1, characterised in that it includes a step to record the new set of HDL description files obtained.

3. (currently amended) Analysis and insertion method according to claim 1 [[or 2]], characterised in that the step (1) for localising HDL instructions at the origin of the memory elements includes:

- a step to search for synchronised processes to detect objects assigned within these processes,
- and application of the following rules for identification of instructions at the origin of memory elements:

- any object assigned within one process and that is read in another process or in the concurrent part of the HDL code will be considered as being a memory element,
- in a synchronised process, any object assigned within one branch of an "if" control structure without being assigned within all other branches of this same structure is considered to be a memory element,
- in a synchronised process, any object that is read before it is written is considered to be a memory element.

4. (currently amended) Analysis and insertion method, according to ~~any one of claims 1 to 3~~ claim 1, characterised in that it includes a step to identify any different existing clock domains, and in that the step to insert HDL memory element chaining instructions is carried out so as to create at least one distinct SCAN chain for each clock domain.

5. (currently amended) Analysis and automatic insertion method according to ~~one of claims 1 to~~ [[4]] claim 1, characterised:

- in that it includes a step (1a) for analysis or indexing of all original HDL description files and creation of at least one indexing file containing the list of design units if they exist (entity, library, packet), for each object and HDL process, and all declarations for each design unit, each declaration including the line number, the object name, type, size and the associated control construction type;
- and in that the step (1b) for localising HDL instructions that will be at the origin of memory elements during synthesis of the circuit, includes a phase to create a memory localisation file comprising, for each memory element, the name of the corresponding HDL object, its type, dimension and coordinates in the original HDL description files.

6. (original) Analysis and automatic insertion method according to claim 5, characterised in that the analysis or indexing step includes an indexing step

of identical instances of the system and in that the step for automatically inserting the SCAN instructions is performed for each instance when it appears for the first time during automatic insertion of SCAN HDL instructions for all description files in the system, then, whenever the said instance is encountered again, a check is made that the SCAN HDL instructions can be used to satisfy local SCAN constraints, and if this is the case, no modification is made to description files of the instance. On the other hand if this is not the case, the description file(s) of the instance is (are) modified and all previous locations in which the said instance occurred are verified and its environment is modified so as to satisfy local SCAN constraints with the new form of the instance.

7. (currently amended) Analysis and automatic insertion method according to ~~one of claims 1 to 5~~ claim 1, characterised in that, in the case of a lack of information related to the dimension of a variable at the origin of a memory element in the original HDL description files, it includes either a step to automatically define this value based on a predetermined default value, or a step in which a user of the method makes the definition interactively.

8. (currently amended) Analysis and automatic insertion method according to ~~one of claims 1 to 7~~ claim 1, characterised in that it includes:

- a step to verify the compatibility of memory elements with each other, when inserting the HDL chaining instructions,
- and in case of incompatibility:
  - either a phase for automatic transformation of the type and / or dimension of one of the two objects at the origin of the conflict,
  - or a phase in which the user interactively modifies the type and / or the dimension of one of the two objects at the origin of the conflict.

9. (currently amended) Analysis and automatic insertion method according to ~~one of claims 1 to 8~~ claim 1, characterised in that the HDL instruction insertion step for chaining of memory elements includes:

- a phase to insert HDL chaining instructions of memory elements, called local chaining, in the set of HDL instructions corresponding to an HDL

object so as to obtain at least one distinct chain of memory elements for each HDL object during the synthesis,

- a phase to insert HDL chaining instructions, called global chaining, in the HDL description files, so as to obtain at least one chain of memory elements during the synthesis, comprising the chains of memory elements created during the local chaining phase.

10. (original) Analysis and automatic insertion method according to claim 9, characterised in that the step for automatic insertion of the HDL instructions for local chaining includes the following phases:

- insertion of HDL instructions corresponding to test signals used as an input–output port,
- insertion of HDL instructions corresponding to intermediate work signals, in the case of memory elements between several processes involving primary input/output ports,
- insertion of HDL instructions in each process, so as to obtain at least one so–called "SCAN" chain during synthesis of the circuit, connecting memory elements specific to the process,
- insertion of HDL instructions providing concurrent assignment of input and output chains of SCAN chains external to processes.

11. (currently amended) Analysis and automatic insertion method according to ~~one of claims 1 to 10~~ claim 1, characterised in that in order to enable reconfiguration of SCAN chains after synthesis, the step to insert HDL SCAN instructions includes:

- a phase to insert HDL instructions that will generate intermediate switches inserted between some at least of the memory elements of a SCAN chain, during synthesis,
- a phase to insert HDL instructions that will generate a controller of these intermediate switches, during synthesis.

12. (currently amended) Analysis and automatic insertion method according to ~~one of claims 1 to 11~~ claim 1, characterised in that the step for automatic insertion of HDL instructions includes a step for insertion of built–in self–

test HDL instructions which will be at the origin of at least the following during the synthesis:

- test pattern generation means such as a test pattern generator (20),
- means of analysing the response of the tested circuit such as a test result compression block (21),
- test control means such as a test controller (22),
- a test input (23) and output (24).

13. (original) Analysis and automatic insertion method according to claim 12, characterised in that the means for generating test patterns include a linear Parallel Random Pattern Generator (PRPG) for which the initialisation sequence is programmable.

14. (currently amended) Analysis and automatic insertion method according to ~~claims 11 and 12 or~~ claim 13, characterised in that means for generating test patterns and analysing responses are based on the reconfigurable SCAN structure.

15. (currently amended) Device for automated design of a complete system or part of an integrated digital electronic system (5) in a description language at the register transfer level, called the HDL language, this device including at least one calculation unit, a memory unit, and a file storage unit, characterised in that the storage unit includes HDL language description files of the system or part of the integrated electronic system and in that the calculation and memory units are adapted to generate new HDL description files of the system or part of this system that include HDL descriptions, using the method according to ~~one of claims 1 to 12~~ claim 1, and starting from HDL description files, such that the system or part of the integrated digital electronic system obtained from the new files includes at least part of the electronic logic circuits necessary for the operational test of the memory elements.

16. (currently amended) Integrated digital electronic system, characterised in that it results from the synthesis of a set of description files in the HDL language obtained by use of the method according to ~~one of claims 1 to 12~~

claim 1 and in that it includes at least part of the logical electronic circuits necessary to test operation of the memory elements, such as one or several SCAN chains.

17. (original) Integrated digital electronic system according to claim 16, characterised in that it is adapted to enable reconfiguration of the SCAN chains and in that it includes at least:

- intermediate switches (15, 16) placed between at least some of the memory elements (11, 12) in a SCAN chain;
- and an intermediate switch controller.

18. (currently amended) Integrated digital electronic system according to claim 16 [[or 17]], characterised in that it includes:

- test pattern generation means such as a test pattern generator (20),
- means of analysing the response of the tested circuit such as a test result compression block (21),
- test control means such as a test controller (22),
- a test input (23) and output (24).

19. (original) Integrated digital electronic system according to claim 18, characterised in that the means for generating test patterns include a linear Parallel Random Pattern Generator (PRPG) for which the initialisation sequence is programmable.

20. (currently amended) Integrated digital electronic system according to ~~claims 17 and 18 or~~ claim 19, characterised in that the means for generating test patterns and analysing responses are based on the reconfigurable SCAN structure.